Asia Pac. J. Math. 2025 **12**:51

HYBRID INTERACTIVE HEURISTIC METHOD BASED ON PFGA AND GOAL PROGRAMMING

BOUREIMA SORY[1], WENDPANGA JACOB YOUGBARÉ[1,*], STÉPHANE AIMÉ METCHEBON TAKOUGANG[2,3]

[1]Laboratory of Mathematics, Computer Science and Applications (L@MIA), University Norbert Zongo , Koudougou, Burkina Faso

[2]Faculty of sciences, New Dawn University, Burkina Faso

[3]LAboratory for Numerical Analysis, Computer Science and Biomathematics (LANIBIO), Joseph Ki-Zerbo University, Ouagadougou, Burkina Faso

[*]Corresponding author: jwyougby@yahoo.com

Received Apr. 16, 2025

Abstract. Meta-heuristics are the most widely used technics in the literature for solving optimization problems and especially combinatorial problems. In the case of a multi-objective problem, such methods construct an approximation of the set of all efficient solutions or Pareto front. For large-scale multi-objective combinatorial problems, the number of efficient solutions may become very large. In order to help a decision maker to make a choice between these solutions, an interactive procedure is developed in this paper. Its principle consists of the use of a meta-heuristic to generate the set of optimal Pareto solutions of the multi-objective problem. The interactive procedure is then applied to obtain the best compromise solution(s). This paper presents a new hybrid interactive heuristic based on the pareto fitness genetic algorithm and goal programming to solve multi-objective optimization problems. The proposed approach integrates the global exploration capabilities of genetic algorithms and the effective management of the multiple objectives of goal programming. An interactive process allows decision-makers to dynamically adjust priorities and preferences, facilitating convergence towards Pareto optimal solutions aligned with specific needs. The effectiveness of this new heuristic is illustrated through four applications.

2020 Mathematics Subject Classification. 90C59; 90C29; 90C27.

Key words and phrases. Multi-objective programming; Combinatorial optimization; Pareto Fitness Genetic Algorithm; Simulated annealing; goal programming; interactive method.

1. Introduction

Nowadays, a company's productivity performance often depends on more than one aspect or objective. This requires a multi-objective approach. These types of optimization problems, although

DOI: 10.28924/APJM/12-51

mathematically poorly posed, most often better reflect the socio-economic reality inherent in optimization problems and are therefore economically well posed. Many real-world applications often require multi-objective combinatorial optimization, because the variables are discrete or binary and the various objectives are usually contradictory but inherent to reality [11].

Multi-objective optimization problems are ubiquitous in various fields such as engineering, management and logistics. The multi-objective optimization problems (MOOP) consist of the simultaneous optimization of several possibly conflicting objective functions.

In the literature, metaheuristics are the most widely used technics for solving these problems. They provide a set of optimal pareto solutions, often of very high size, requiring alternative approaches to direct the decision-maker towards a limited set of optimal pareto solutions containing the "most preferred" solution for the decision-maker also known as the "best compromise solution". Some commonly used approaches are hierarchical methods, weighted sum methods, goal-programming methods, interactive methods, etc.

- In the hierarchical approach, objectives are prioritized and optimization is done in that order.
- The weighted sum approach uses a function of aggregation to group the objectives considered into an overall objective.
- The goal-programming approach allows for multiple goals to be managed by converting them into goals with pre-defined target values and weights. The weights reflect the relative importance or priority of each objective.
- In the interactive approach, the principle is to collect prior to each step of the interaction, information from the decision-maker on his preferences for the proposed solution so that gradually the method converges towards a compromise satisfying the intended objective.

The interactive approach has been the subject of many studies in the literature and draws our attention in this paper. Yougbaré (2019) [18] has developed an interactive method based on the goal-programming method and the Data Envelopment Analysis approach. The model was tested on a resource allocation problem.

Work such as [9,11,14], focused on an interactive model based on the multi-objective simulated annealing (MOSA) method which is an adaptation of simulated annealing to the multi-objective framework. The model has been tested on several production problems such as the multi-objective knapsack problem and the assignment problem. Formally, given a set of feasable solutions $D \subseteq \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^K$ $K \geq 2$ objective functions, the multi-objective optimization problem is written:

$$(P) \quad \max \{f(x) : x \in D\} \tag{1}$$

The aim of this work is to develop an interactive multi-objective approach with a generalist status whose process relies on an optimal Pareto set provided by a metaheuristic based on Pareto Fitness

Genetic Algorithm (PFGA). This approach should cover problems such as:

$$\begin{aligned} \max f(x) = \quad & (f_1(x), \ldots, f_K(x)) \text{ where } x \in D \\ D = \quad & \{x \in \mathbb{R}^n : g_s(x) \leq 0, \quad s = 1, \ldots, m\}. \end{aligned} \tag{2}$$

A solution $x^* \in D$ is efficient for (P) if there does not exist any other solution $x \in D$ such that: $f_k(x) \geq f_k(x*), k = 1, \ldots, K$ with at least one strict inequality. Then, the vector $f(x*)$, of values $f_k(x*), k = 1, \ldots, K$ is said to be non dominated in the space of objective functions. Let $E(P)$ denote the set of all efficient solutions to a problem P. The Pareto front $f(E(P))$ is the subset of $f(D)$ containing Pareto optimal solutions or non-dominated solutions. It helps the decision maker for identifying the best compromise solutions.

In mathematics, metaheuristics are known for their effectiveness in solving MOOP. In the literature, many metaheuristics exist but the universal metaheuristic to effectively solve any type of multi-objective problem does not exist. There are most known existing methods solving MOOP but the evolutionary algorithms seem to be particularly suitable for MOOP [9,12,15,20].

These problems often involve conflicting objectives requiring delicate compromises. Pareto fitness genetic algorithms (PFGA) are known for their ability to effectively explore complex solution spaces by identifying approximate Pareto fronts. At the same time, goal programming provides a structured framework for integrating and prioritizing multiple objectives while minimizing deviations from targets. However, these methods have limitations when used in isolation, particularly in terms of convergence and taking into account the preferences of decision makers.

Several authors have been interested in the implementation of hybrid interactive methods based on "goal programming" (GP) and/or genetic algorithms such as [2,3,10,16].
To overcome these limitations, we propose a hybrid interactive heuristic combining the strengths of PFGA and GP. This approach aims to exploit the global exploration capacity of genetic algorithms while integrating the flexibility and structuring of programming by objectives. Interactivity is at the heart of this heuristic, allowing decision-makers to dynamically adjust priorities and preferences based on intermediate outcomes, thus promoting convergence towards optimal solutions better aligned with specific needs.

2. New Method Based on Pareto Fitness Genetic Algorithm Using Polynomial Mutation and Logarithmic Mean

This section describes a new meta heuristic used to generate the set of Pareto optimal solutions. It is a new heuristic based on the Pareto Fitness Genetic Algorithm (PFGA) using polynomial mutation and the logarithmic mean that we will call PFGA-LM. We will describe the main steps of this method.

PFGA-LM is an adaptation of the Pareto Fitness Genetic Algorithm (PFGA) for solving multi-objective optimization problems. We propose to include polynomial mutation and to modify the double ranking strategy proposed by Elaoud et al.(2007) [7] in order to improve the performance.

2.1. **Algorithm PFGA-LM.** The algorithm of PFGA-LM is summarized by the following steps:

Step 1: : Initializing at random a population of solutions

Step 2: While (non-Stop): – Evaluate the current population;

– Update the elitist set by coping non-dominated solutions; – Assigning rank (R) and density value to each individual;

– Calculate the corresponding fitness;

– Set $i \Leftarrow 0$;

– While $(i < R)$: Select a solution randomly from the elitist external set, Reintroduce the selected solution to the next population, $i \Leftarrow i + 1$;

– While $(i < Population\ size)$: Select two different parents, Perform crossover, Introduce children into the next population, $i = i + 2$, Perform mutation on one randomly selected child.

2.2. **Assigning rank.** The double ranking strategy is to assign to x_i a shadow rank value $R'(x_i)$ representing the number of solutions x_j that dominate x_i in the current population denoted P_c. Formally, $R'(x_i)$ is obtained by:

$$R'(x_i) = |x_j \in P_c / x_i \prec x_j|, \quad \forall x_i \in P_c. \tag{3}$$

where $x_i \prec x_j$ means that x_j dominates x_i.

An individual's rank $R(x_i)$ is then defined by the following relation :

$$R(x_i) = R'(x_i) + \sum_{x_j \in P_c / x_i \prec x_j} R'(x_j), \quad \forall x_i \in P_c. \tag{4}$$

In this paper, we use the double ranking strategy proposed in [7].

2.3. **Population adaptation density.** The principle of the population adaptation density estimation strategy consists in cutting the decision space into cells of identical sizes for each objective. We propose a new strategy than the one proposed in [7]. Elaoud et al. [7], refer to a population where each individual occupies $1/N$ of the objective space measure. The cell width Wd_i in each objective dimension i may be calculated with the following formula 5:

$$Wd_i = \frac{f_i^M - f_i^m}{\sqrt[k]{N}}, i = 1, \ldots, k. \tag{5}$$

where $f_i^M = \max_{x \in X} f_i(x)$ and $f_i^m = \min_{x \in X} f_i(x)$.

This case is optimal for the objective space where the objective functions are linear. For non-linear objectives, the choice of cell size should take into account other principles. To remedy this, we have proposed another choice of cell size in case $f_i^m > 0, i = 1, \ldots, k$ by the following formula 6:

$$Wd_i = M_{ln}\left(f_i^m, f_i^M\right), i = 1, \ldots, k. \tag{6}$$

where

$$M_{ln}(a,b) = \begin{cases} a, if \quad a = b \\ \frac{b-a}{\ln b - \ln a} \quad otherwise. \end{cases} \tag{7}$$

for $a > 0, b > 0$ and $\ln$ designs the logarithm base e. Our principle takes into account the fact that:

- on the one hand, it underestimates the actual distance between the points corresponding to f_i^M et f_i^m. It corresponds to the deviation of the orthogonal projections of those points. Therefore, cell size is in turn underestimate;
- on the other hand, the distribution of the solutions obtained by a method is evaluated on the Pareto front and not on the axes of objective functions. Thus, the distance between two points corresponds to that of these points on the Pareto front and not on the axes of the objective functions.

We denote n_{ci} the number of cells for f_i and n_{ci} is defined by:

(1) if $f_i^M - f_i^m \leq Wd_i$ we take $nc_i = 1$;
(2) otherwise let $\dfrac{1}{\ln\left(f_i^M\right) - \ln\left(f_i^m\right)} = a_i + r_i$ such that a_i is integer and $r_i \in [0,1]$ we take

$$nc_i = \begin{cases} a_i \quad if \quad r_i = 0, \\ a_i + 1 \quad otherwise. \end{cases}$$

2.4. **Fitness assignment.** The fitness assignment proposed in [7] is used and the fitness of an individual is

$$f(x_i) = \frac{1}{exp(R(x_i)) * D(x_i)}, \tag{8}$$

where $D(x_i)$ denotes the density value of solution x_i. If x_i is a non-dominated solution, $R(x_i) = 0$ and so $exp(R(x_i)) = 1$.

Selection. In this paper, we use the selection method proposed in [7]. The authors propose a binary stochastic sampling without replacement, which selects two different parents from the current population as follows:

- Summing the fitness of all population members having an acceptable rank (we name this summation as the total fitness).
- Normalizing the fitness of each considered individual by dividing it by the total fitness.
- Generate a random number (R1) between 0 and 1.

- The first selected individual i is whose normalized fitness, added to the preceding individual, is greater or equal to (R1).
- Generate a second random number (R2) between 0 and 1 private from the interval between the added normalized fitness of the individual $i-1$ and the added normalized fitness of the individual i.
- The second selected individual is whose normalized fitness, added to the preceding individual, is greater or equal to (R2).

The main particularity of this method is that it is based on polynomial mutation.

2.5. **Update the elitist set.** Just like in [6,7], a set is created and updated at each step to maintain elitist solutions in the algorithm process. This set gives at the end the final Pareto front.
During the evolutionary process, a small random number R ($R = 1$ or 2) of elitist external set solutions are also randomly selected.

2.6. **Polynomial mutation.** During the evolution process of a genetic algorithm, there are several candidates of mutation types. It should be noted that the type of mutation used is not specified in [7]. Given that the polynomial mutation testifies to the precision and stability for the methods that use it and that for a significant number of problems [1,8], we opted for the polynomial mutation in this method.
Considering x a solution to be muted, r an actual number such as $r \in [0;1]$:

- x_l lower bound of x ;
- x_u upper bound of x ;
- p_m the probability of mutation of the algorithm ;
- $\delta_1 = \dfrac{x - x_l}{x_u - x_l}$, the standard variance (normalized) of the x solution with its lower bound x_l ;
- $\delta_2 = \dfrac{x_u - x}{x_u - x_l}$, the standard variance (normalized) of the x solution with its upper bound x_u ;
- η a natural integer denoting the distribution index.

2.7. **Approximation of the optimal pareto front.** We propose to include polynomial mutation and to modify the double ranking strategy proposed in [7] into the Pareto Fitness Genetic Algorithm (PFGA) in order to improve the performance.

3. New Heuristic for Multi-Objective Optimization Problem

In this section, we describe the different phases of the new heuristic Multi-Objective Optimization Problem (H-MOOP)

3.1. **initialization.** We start by determining an approximation of the optimal Pareto front. From this optimal Pareto front, the nadir point and the ideal point are determined. These points are determined from the optimal Pareto front.

3.2. **Determination of the goal solution.** The method we have developed is also based on the work of Yougbaré in [18] and Teghem et al. in [14].

- Yougbaré in [18] has developed an interactive model based on the goal-programming method using jointly the multi-criteria decision aid character of the DEA (Data Envelopment Analysis) method. The model was tested on a resource allocation problem;
- Teghem et al. [14] have developed an interactive heuristic based on themulti-objective simulated annealing (MOSA) method. This method was then effectively tested on several production problems such as the multi-objective backpack problem, the multi-objective assignment problem and the multi-objective scheduling problem.

The goal solution is a kind of compromise solution that takes into account the aspirations and judgments of the decision-maker in relation to the various criteria, taking into account the ideal and nadir points of the multi-objective problem. We will recall the model 9 proposed by Yougbaré in [18] to characterize a target solution. This characterization takes into account the information provided by the decision-maker and a subset of satisfactory optimal Pareto solutions.

$$\begin{cases} max_{s\in E(P)} = \sum_{r_d\in \mathbf{K}^d} \gamma_{r_d}(s) + \epsilon \sum_{r\in \mathbf{K}^d} t_r(s) & \\ \gamma_{r_d}(s) y_{r_d,d} - \sum_{j=1}^{J+1} \delta_j y_{r_d,j} = 0, & \forall r_d \in \mathbf{K}^d \\ \sum_{j=1}^{J+1} \delta_j y_{r,j} - t_r = y_{r,d} & \forall r \notin \mathbf{K}^d \\ \sum_{j=1}^{J+1} \delta_j = 1, \quad \delta_j \geq 0, \quad j = 1; ...; J & \\ \gamma_{r_d} > 1, & \forall r_d \in \mathbf{K}^d \\ t_r \geq 0 & \forall r \notin \mathbf{K}^d \end{cases} \tag{9}$$

where

- $J = |E(P)|$;
- $\epsilon > 0$ is positive real small enough ;
- δ_j refers to the importance of the jth solution j ;
- γ and t respectively designate directional factors of the research for the objectives concerned by the improvement and for those not concerned ;
- y is the common solution so a part of the set $f(E(P))$;
- y_{r_d} is the value of y for the objective f_{r_d}, $r_d \in \mathbf{K}^d$;
- y_r is the value of y for the objective f_r, $r \notin \mathbf{K}^d$;
- $y_{r_d,j}$ s the value of the jth component of the vector f_{r_d}.

We are adapting the model 9 to obtain a new model 10 following:

$$\begin{cases} \text{q=} & max_{y\in L_m} \sum_{r_d\in \mathbf{K}^d} \gamma_{r_d}(y) \\ & \gamma_{r_d}(y)y_{r_d,d} - \dfrac{1}{J+1}\sum_{j=1}^{J+1} y_{r_d,j} = 0, \quad \forall r_d \in \mathbf{K}^d \\ & \gamma_{r_d} > 1, \qquad\qquad\qquad\qquad\qquad \forall r_d \in \mathbf{K}^d \end{cases} \tag{10}$$

The goal solution g of the multi-objective model is defined as:

$$\begin{gathered} g_{r_d} = q \times y_{r_d}, \forall r_d \in \mathbf{K}^d \\ g_r = y_r, \forall r \notin \mathbf{K}^d \end{gathered} \tag{11}$$

where $\mathbf{K}^d$ is the set of indixes of objectives to be improved.

3.3. **Algorithm H-MOOP.**

3.3.1. *Principle.* The principle of this new interactive multi-objective heuristic is based on the generation of a set of effective solutions by PFGA-LM. This set is an approximation of the complete set of effective solutions. Once this set is generated, it is restricted as the decision-maker reacts to the different solutions proposed in the various stages of the iteration process. The procedure stops when the decision-maker is satisfied or when this package no longer contains a solution, that is to say, it can no longer improve the performance of the current solution. It applies specifically to maximization problems. When it is a minimization problem, a transformation is necessary to return to a maximization problem. This is the mind used by the DEA method for the case of undesirable outputs.

3.3.2. *Steps of the algorithm.* The H-MOOP algorithm can be summarized by the following steps

: Step 1: initializing Generate $E(P)$ by PFGA-LM method. Take $m = 0, L_m = E(P)$. Randomly select $z \in f(E(P))$.

: Step 2: Iterative process and interaction with the decision maker While the decision maker is not satisfied: Updating K^d through interaction with the decision maker; Calculate g; Updating L_m $L_m = L_{m-1} \bigcap \{x \in E(P) | y_{r_d} \geq g_{r_d}, y = f(x)\}$; Resume process with $m = m + 1$.

4. Applications

We will apply the H-MOOP method to three examples: an example of a multiobjective knapsack problem (maximization), a non-linear multiobjective test problem (maximization) and a multiobjective linear minimization problem. In any case we will consider a fictitious decision-maker who will interact in the interactive process. Note Y_m the performance vector proposed to the decision-maker at the iteration m, f_{r_d} the objective chosen for improvement, L_m the set of selectable performance vectors at the iteration m and $|L_m|$ its cardinal, γ_m the optimal solution of the system (2) at the iteration m, and g_m the goal at the iteration m.

4.1. **Application 1.** We apply our method to a multiobjective combinatorial optimization problem, namely the multi-objective knapsack problem formulated as follows [14]:

$$\begin{cases} max \quad z_k(X) = \sum_{j=1}^{n} c_j^k x_j, \quad k = 1, ..., K \\ \sum_{j=1}^{n} w_j x_j \leq W \\ w_j \leq W, \quad j = 1, ..., n \\ \sum_{j=1}^{n} w_j > W, \quad k = 1, ..., K \\ x_j \in \{0; 1\}, \quad j = 1, ..., n \end{cases} \tag{12}$$

To solve the problem (4), we consider the following hypotheses.

- $c_j^k \in [1; 50]$, $j = 1; ...; n$, $k = 1; ...; K$;
- $\overline{z_k} = \dfrac{max(z_k) + min(z_k)}{2}$; $k = 1; ...; K$;
- The weights of objects w_j are randomly selected in $[1; 50]$;
- The knapsack W capacity is selected as follows: $\dfrac{W}{\sum_{j=1}^{n} w_j} = 0.5$;
- $n = 50000$: the number of objects or the size of the problem;
- we take $K = 2$.

The results obtained from the different iterations are summarised in the following table 1:

TABLE 1. *Summary of the application of H-MOOP to example 1*

Iteration m	$\|E(P)\|$	Y_m	z	γ_m	g_m
1	7	(65773,65792)	z_2	1.0040	(66279;65277)
2	3	(66062,65367)	z_2	1.0042	(66062;65644)
3	2	(64717,66051)	z_1	1.0054	(65069;66051)
4	1	(64717,66051)	z_2	1	(65773;65792)

The efficient performance solution $(64717, 66051)$ is the best compromise solution at the end of the procedure.

4.2. **Application 2.** In this example, we consider six (06) objectives that is to say we take $K = 6$.

The results obtained from the different iterations are summarized in the following table 2:

TABLE 2. *Summary of the application of H-MOOP to example 2*

Iteration m	$\|L_m\|$	y_m	fr_d	γ_m	g_m
1	73	(65728, 65671 , 66802, 66037, 66440, 65385)	z_4	1.0063	(65728, 65671, 66802, 66455, 66440, 65385)
2	35	(65739, 65516, 66495, 66692, 65722, 65609)	z_5	1.0032	(65739, 65516, 66495, 66692, 65932, 65609)
3	26	(64997, 66219, 66399, 67521, 66279, 65442)	z_5	1.0006	(64997, 66219, 66399, 67521, 66316, 65442)
4	19	(65362, 65576, 66873, 66526, 66837, 65921)	z_2	1.0006	(65362, 65617, 66873, 66526, 66837, 65921)
5	6	(65594, 65986, 65807, 67617, 66790, 65144)	z_2	1.0032	(65594, 66200, 65807, 67617, 66790, 65144)
6	2	(65862, 65299, 66144, 66761, 66533, 65715)	z_2	1.0065	(65865, 66925, 66091, 66654, 67629, 64634)
7	0	–	–	–	–

The efficient performance solution $(65865, 66925, 66091, 66654, 67629, 64634)$ is the best compromise solution at the end of the procedure.

4.3. **Application 3.** The purpose of this problem is to minimize simultaneously (z_1) and (z_2). The mathematical formulation of the problem leads to a bi-objective optimization problem as presented below.

$$\begin{cases} min \quad z_1(x) = 0.1x_1 + 1.3x_2 + 0.3x_3 + 0.42x_4 + 0.8x_5 + 0.3x_6 + \\ \qquad 0.5x_7 + 1.5x_9 + 0.62x_{10} + 0.12x_{11} + 1.0x_{12} + 0.5x_{13} + \\ \qquad 0.74x_{15} + 0.24x_{16} + 1.62x_{17} + 1.12x_{18} + 0.62x_{19} + \\ \qquad 0.12x_{20} + 2.0x_{21} + 1.5x_{22} + 1.0x_{23} + 0.5x_{24} \\ min \quad z_2(x) = 2x_1 + 1x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 2x_7 + \\ \qquad 2x_8 + 1x_9 + 2x_{10} + 1x_{11} + 2x_{12} + 3x_{13} + 3x_{14} + \\ \qquad 2x_{15} + 3x_{16} + 1x_{17} + 2x_{18} + 3x_{19} + 4x_{20} + 1x_{21} + \\ \qquad 2x_{22} + 3x_{23} + 4x_{24} + 4x_{25} \\ s.c \begin{cases} 2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 10 \\ x_3 + 2x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} \geq 15 \\ x_4 + x_{10} + x_{11} + 2x_{15} + 2x_{16} + x_{17} + x_{18} + x_{19} + x_{20} \geq 30 \\ x_5 + 2x_6 + x_8 + x_{11} + x_{12} + 2x_{13} + 3x_{14} + x_{16} + x_{18} \\ +2x_{19} + 3x_{20} + x_{21} + 2x_{22} + 3x_{23} + 4x_{24} + 5x_{25} \geq 8 \\ x_i \geq 0, \quad i \in \{1; ...; 25\} \end{cases} \end{cases} \tag{13}$$

The results obtained from the different iterations are summarized in the following table 3:

TABLE 3. *Summary of the application of H-MOOP to example 3*

Iteration m	$\lvert L_m \rvert$	y_m	f_{r_d}	γ_m	g_m
1	10	$(18.5000; 99.0000)$	f_2	1.0514	$(18.5000; 104.0909)$
2	5	$(19.7000; 93.0000)$	f_1	1.0555	$(20.7933; 93.0000)$
3	3	$(19.7000; 93.0000)$	f_1	1.0081	$(19.8589; 93.0000)$
4	2	$(18.5000; 99.0000)$	f_1	1.0216	$(18.9000; 99.0000)$
5	1	–	–	–	–

The efficient performance solution $(18.5000; 99.0000)$ is the best compromise solution at the end of the procedure.

4.4. **Application 4.** We consider a non-linear bi-criterion maximization problem as follows:

$$\begin{cases} max \quad z_1(x) = 1.1 - x_1 \\ max \quad z_2(x) = 60 - \dfrac{1 - x_2}{x_1} \\ \qquad x = (x_1; x_2) = [0.1; 1] \times [0; 5] \end{cases} \tag{14}$$

The results obtained from the different iterations are summarized in the following table 4:

TABLE 4. *ummary of the application of H-MOOP to example 4*

Iteration m	$\|L_m\|$	y_m	f_{r_d}	γ_m	g_m
1	200	$(0.2130; 58.8726)$	z_1	3.2231	$(0.6865; 58.8726)$
2	115	$(0.8623; 55.7932)$	z_1	1.0368	$(0.8940; 55.7932)$
3	65	$(0.9467; 53.4766)$	z_2	1.0152	$(0.9467; 54.2900)$
4	10	$(0.9031; 54.9208)$	z_1	1.0091	$(0.9113; 54.9208)$
5	5	$(0.9203; 54.4339)$	z_2	1.0002	$(0.9203; 54.4474)$
6	2	$(0.9139; 54.6276)$	z_2	1.0015	$(0.9139; 54.7071)$
7	0	–	–	–	–

The efficient performance solution $(0.9139; 54.6276)$ is the best compromise solution at the end of the procedure.

5. Conclusion and Future Work

The hybrid interactive heuristic presented in this paper is a significant step forward in the field of multi-objective optimization. By combining the advantages of Pareto fitness genetic algorithms and goal-programming, this approach offers increased flexibility and efficiency in solving complex problems. The integration of an interactive process allows active involvement of decision-makers, ensuring that the solutions obtained accurately reflect their preferences and priorities. Four applications illustrate the potential and versatility of this heuristic. Future research could explore the application of this approach to other areas and study its comparative effectiveness with other hybrid methods.

Abbreviations.

DM: Decision Maker

GP: Goal Programming

H-MOOP: heuristic Multi-Objective Optimization Problem

MOOP: Multi-Objective Optimization Problem

PFGA: Pareto Fitness Genetic Algorithm

PFGA-LM: Pareto Fitness Genetic Algorithm with polynomial mutation and algorithmic mean.

Acknowledgments. The authors thank also the anonymous referees who helped to reinforce the scientific quality of this paper by their relevant critics.

Author Contributions.

Boureima Sory: Implementation of the models, writing

Wendpanga Jacob Yougbaré: Conception of the work, methodology, writing

Stéphane Aimé Metchebon Takougang: validation, writing

Conflicts of Interest. The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] B.H. Abed-alguni, Island-Based Cuckoo Search With Highly Disruptive Polynomial Mutation, Int. J. Artif. Intell. 17 (2019), 57–82.

[2] B. Chaudhuri, R. Jana, D.K. Sharma, P. Dan, A Hybrid Genetic-goal Programming Approach for Improving Group Performance in Cell Formation Problems, Int. J. Adv. Oper. Manag. 12 (2020), 377–395. https://doi.org/10.1504/ijaom.2020.112734.

[3] S. Dhouib, A. Kharrat, H. Chabchoub, Goal Programming Using Multiple Objective Hybrid Metaheuristic Algorithm, J. Oper. Res. Soc. 62 (2011), 677–689. https://doi.org/10.1057/jors.2009.181.

[4] M. Díaz-Madroñero, D. Peidro, P. Vasant, Vendor Selection Problem by Using an Interactive Fuzzy Multi-objective Approach with Modified S-curve Membership Functions, Comput. Math. Appl. 60 (2010), 1038–1048. https://doi.org/10.1016/j.camwa.2010.03.060.

[5] K. Deb, R.B. Agrawal, Simulated Binary Crossover for Continuous Search Space, Complex Syst. 9 (1995), 115–148.

[6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002), 182–197. https://doi.org/10.1109/4235.996017.

[7] S. Elaoud, T. Loukil, J. Teghem, The Pareto Fitness Genetic Algorithm: Test Function Study, Eur. J. Oper. Res. 177 (2007), 1703-1719. https://doi.org/10.1016/j.ejor.2005.10.018.

[8] M. Hamdan, On the Disruption-Level of Polynomial Mutation for Evolutionary Multi-Objective Optimisation Algorithms, Comput. Inform. 29 (2010), 783–800.

[9] T. Loukil, J. Teghem, P. Fortemps, A Multi-objective Production Scheduling Case Study Solved by Simulated Annealing, Eur. J. Oper. Res. 179 (2007), 709–722. https://doi.org/10.1016/j.ejor.2005.03.073.

[10] A.O. Mogbojuri, O.A. Olanrewaju, Goal Programming and Genetic Algorithm in Multiple Objective Optimization Model for Project Portfolio Selection: A Review, Nigerian J. Technol. 41 (2022), 862–869. https://doi.org/10.4314/njt.v41i5.6.

[11] T. Loukil, J. Teghem, D. Tuyttens, Solving Multi-objective Production Scheduling Problems Using Metaheuristics, Eur. J. Oper. Res. 161 (2005), 42–61. https://doi.org/10.1016/j.ejor.2003.08.029.

[12] R. Sarker, K. Liang, C. Newton, A New Multiobjective Evolutionary Algorithm, Eur. J. Oper. Res. 140 (2002), 12–23. https://doi.org/10.1016/s0377-2217(01)00190-4.

[13] O. Schutze, X. Esquivel, A. Lara, C.A.C. Coello, Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization, IEEE Trans. Evol. Comput. 16 (2012), 504–522. https://doi.org/10.1109/tevc.2011.2161872.

[14] J. Teghem, D. Tuyttens, E. Ulungu, An Interactive Heuristic Method for Multi-objective Combinatorial Optimization, Comput. Oper. Res. 27 (2000), 621–634. https://doi.org/10.1016/s0305-0548(99)00109-4.

[15] Y. Tian, R. Cheng, X. Zhang, M. Li, Y. Jin, Diversity Assessment of Multi-objective Evolutionary Algorithms: Performance Metric and Benchmark Problems [Research Frontier], IEEE Comput. Intell. Mag. 14 (2019), 61–74. https://doi.org/10.1109/mci.2019.2919398.

[16] N. Wichapa, P. Khokhajaikiat, Using the Hybrid Fuzzy Goal Programming Model and Hybrid Genetic Algorithm to Solve a Multi-objective Location Routing Problem for Infectious Waste Disposal, J. Ind. Eng. Manag. 10 (2017), 853–886. https://doi.org/10.3926/jiem.2353.

[17] J.W. Yougbaré, J. Teghem, Relationships Between Pareto Optimality in Multi-objective 0–1 Linear Programming and Dea Efficiency, Eur. J. Oper. Res. 183 (2007), 608–617. https://doi.org/10.1016/j.ejor.2006.10.026.

[18] J.W. Yougbaré, Interactive Dea-goal Programming Method, Far East Journal of Mathematical Sciences (FJMS) 103 (2018), 573–586. https://doi.org/10.17654/ms103030573.

[19] J.W. Yougbare, Multicriteria Allocation Model of Additional Resources Based on DEA: MOILP-DEA, Iran. J. Numer. Anal. Optim. 9 (2019), 193–205. https://doi.org/10.22067/ijnao.v9i2.81034.

[20] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evol. Comput. 8 (2000), 173–195. https://doi.org/10.1162/106365600568202.